ETSI TS 103 096-3 V1.1.1 (2013-07)

Intelligent Transport Systems (ITS);
Testing;
Conformance test specification for TS 102 867 and TS 102 941;
Part 3: Abstract Test Suite (ATS) and Protocol Implementation eXtra Information for Testing (PIXIT)

Reference
DTS/ITS-0050020

Keywords
ATS, ITS, security, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 3 of a multi-part deliverable covering Conformance test specification for ITS Security as identified below:

TS 103 096-1: "Protocol Implementation Conformance Statement (PICS)";

TS 103 096-2: "Test Suite Structure and Test Purposes (TSS&TP)";

TS 103 096-3: "Abstract Test Suite (ATS) and Protocol Implementation eXtra Information for Testing (PIXIT)";

TR 103 096-4: "Validation report".

1 Scope

The present document provides parts of the Abstract Test Suite (ATS) for Security as defined in IEEE 1609.2 [1], TS 102 941 [2] and in TS 102 867 [3] in compliance with the relevant requirements and in accordance with the relevant guidance given in ISO/IEC 9646-7 [8]. The TTCN modules and PIXIT declarations are not defined in the present document and will be added in a later revision.

The ISO standard for the methodology of conformance testing (ISO/IEC 9646-1 [5] and ISO/IEC 9646-2 [6]) as well as the ETSI rules for conformance testing (ETS 300 406 [9]) are used as a basis for the test methodology.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] IEEE P1609.2/D12 (January 2012): "IEEE Draft Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages".

[2] ETSI TS 102 941: "Intelligent Transport Systems (ITS); Security; Trust and Privacy Management".

[3] ETSI TS 102 867: "Intelligent Transport Systems (ITS); Security; Stage 3 mapping for IEEE 1609.2".

[4] ETSI TS 103 096-1: Conformance test specification for Security Test requirements and Protocol Implementation Conformance Statement (PICS) proforma.

[5] ISO/IEC 9646-1 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework - Part 1: General concepts".

[6] ISO/IEC 9646-2 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 2: Abstract Test Suite specification".

[7] ISO/IEC 9646-6 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 6: Protocol profile test specification".

[8] ISO/IEC 9646-7 (1995): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 7: Implementation Conformance Statements".

[9] ETSI ETS 300 406 (1995): "Methods for testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[10] FIPS PUB 186-3: "Digital Signature Standard (DSS)".

[11] SEC 1: "Elliptic Curve Cryptography".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI EG 202 798: "Intelligent Transport Systems (ITS); Testing; Framework for conformance and interoperability testing".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms given in IEEE 1609.2 [1] TS 102 941 [2], TS 102 867 [3], ISO/IEC 9646-6 [7] and ISO/IEC 9646-7 [8] apply.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AA	Authorization Authority
ASP	Abstract Service Primitive
ATM	Abstract Test Method
ATS	Abstract Test Suite
CA	Certification Authority
EA	Enrolment Authority
EB	Exceptional behaviour
ITS	Intelligent Transport System
ITS-S	ITS Station
IUT	Implementation Under Test
LDM	Local Dynamic Map
MTC	Main Test Component
PCTR	Protocol Conformance Test Report
PSID	Provider Service Identifier
SAP	Service Access Point
SCS	System Conformance Statement
SCTR	Static Conformance Test Report
SUT	System Under Test
TP	Test Purposes
TSS	Test Suite Structure
TTCN	Testing and Test Control Notation

4 Abstract Test Method

This clause describes the ATM used to test the ITS-Security framework.

4.1 Abstract protocol tester

The abstract protocol tester used by the ITS-Security test suite is described in figure 1. The test system will simulate valid and invalid protocol behaviour, and will analyze the reaction of the IUT.

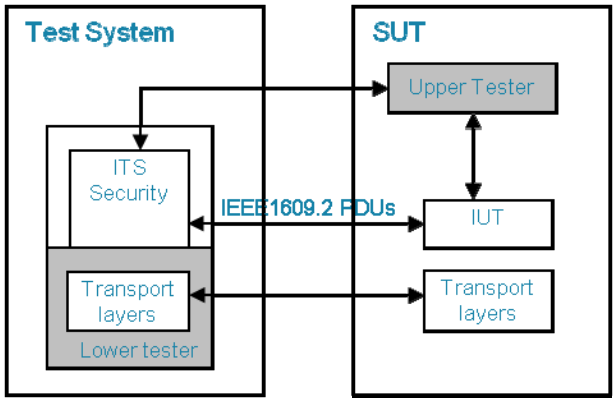

Figure 1: Abstract protocol tester - Security

4.2 Test Configuration

This test suite uses four test configurations in order to cover the different test scenarios. In these configurations, the tester simulates one or several ITS station implementing the ITS Security framework.

4.2.1 Test configuration CF01

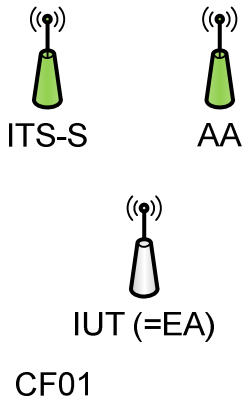

Figure 2: Test Configuration 1

4.2.2 Test configuration CF02

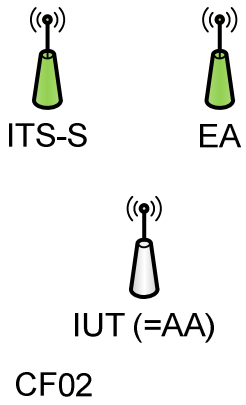

Figure 3: Test Configuration 2

4.2.3 Test configuration CF03

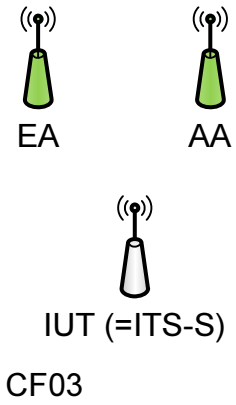

Figure 4: Test Configuration 3

4.2.4 Test configuration CF04

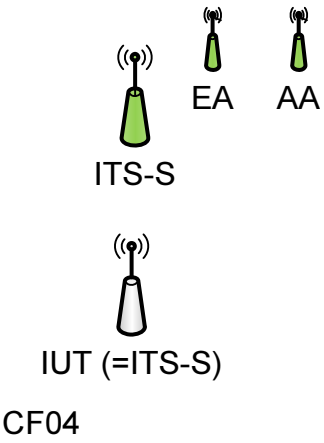

Figure 5: Test Configuration 4

4.3 Test architecture

The present document implements the general TTCN-3 test architecture described in EG 202 798 [i.1], clauses 6.3.2 and 8.3.1.

Figure 6 shows the TTCN-3 test architecture used for the ITS-Security ATS. In single-component testcases (configuration CF04), the MTC is of type ItsSec and communicates with the IUT over securityPort. In multi-component testcases (configuration CF01, CF02 and CF03), the MTC is of type ItsMtc and is used to synchronize the different PTCs. The PTCs are implemented using ItsSec components and communicate with the IUT over securityPort. Port securityPort is used to exchange IEEE 1609.2 [1] protocol messages between the Security test components and the IUT.

The Upper tester entity in the SUT enables triggering Security related functionalities by simulating primitives from application or LDM entities. It is required to trigger the ITS-Security layer in the SUT to send facility messages, which are resulting from upper layer primitives. Furthermore, receiving secured messages may result in the ITS-Security layer sending primitives to the appropriate facility layer.

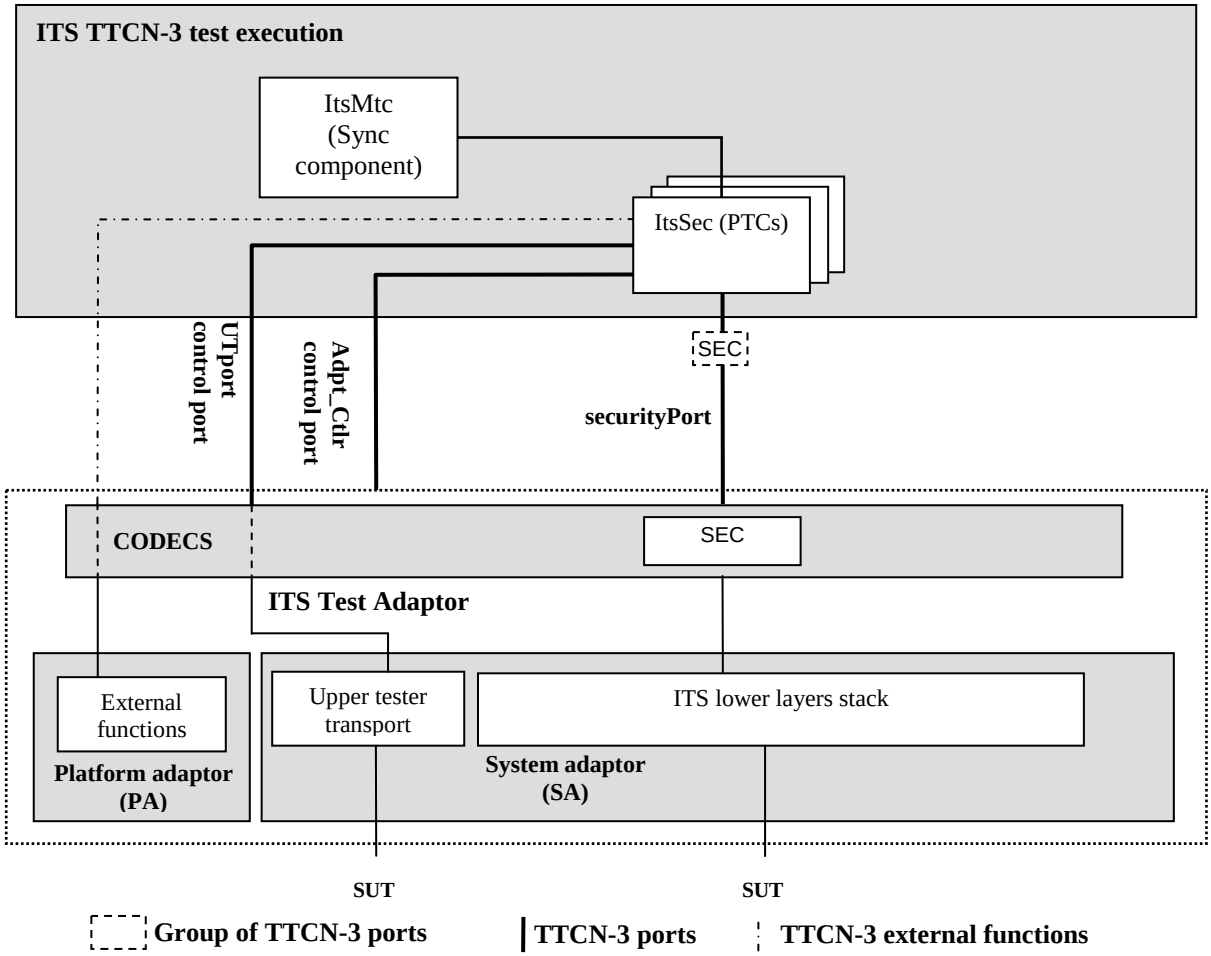

Figure 6: test system architecture

4.4 Ports and ASPs

Two ports are used by the ITS-Security ATS:

- The securityPort, of type SecurityPort
- The utPort of type UpperTesterPort

4.4.1 Primitives of the securityPort

Two types of primitives are used in the securityPort:

- The ieee1609Dot2Ind primitive used to receive messages of type Ieee1609Dot2Message.
- The ieee1609Dot2Req primitive used to send messages of type Ieee1609Dot2Message.

4.4.2 Primitives of the utPort

This port uses three types of primitives:

- The UtInitialize primitive used to initialise IUT
- The UtConfigure primitive used to configure specific options in IUT
- The UtTrigger primitive used trigger actions in IUT
- The UtStatus primitive used to retrieve status information from IUT

Table 1 lists all configuration options that the Test system should be able to modify in IUT for correct test execution.

Table 1: IUT configuration options

Configuration options
use certificate_chain
use explicit_certificates
use a self-signed enrolment request
use start_validity flag and not a lifetime_is_duration
use use_start_validity and lifetime_is_duration
use sec_data_exch_identified_localized
use signature of form x_coordinate_only (use of FIPS 186-3 specification [10])
use compressed public keys in signature (SEC 1 specification [11])
use uncompressed public keys in signature
include generation time when signing a message
include expiry_time when signing a message
include generation _location when signing a message
use ecdsa_nistp256_with_sha256 as PKAlgorithm when signing a message
put certificate in each of the signed messages

Table 2 summarizes the actions that the Test System may require the IUT to perform via UtTrigger primitive.

Table 2: Actions triggered in IUT

Actions
send an EnrolmentRequest message
send an EnrolmentRequest message with more than 8 PSID records
send an AuthorizationRequest message
send a signed message
send a signed message with partial data
send a signed message with external data
send multiple signed message

Status codes returned by IUT via UtStatus primitive upon receipt of a IEEE 1609.2 [1] message are listed in table 3.

Table 3: IUT status codes

Status codes
ACCEPTED
DISCARDED
REJ_INVALID_REGION
REJ_INVALID_VALIDITY_PERIOD
REJ_INVALID_PERMISSIONS
REJ_UNSUPPORTED_SIGNER_TYPE
REJ_INVALID_EXPIRATION
REJ_DUPLICATED_PERMISSIONS
REJ_FORBIDDEN_SUBJECT_TYPE
REJ_FORBIDDEN_CF
REJ_FORBIDDEN_PERMISSIONS
REJ_FORBIDDEN_ACK
REJ_INVALID_REQUESTED_HASH
REJ_EXPIRED_DATA
REJ_IRRELEVANT_REGION
REJ_REVOKED_CERTIFICATE
REJ_UNAUTHORIZED_REGION
REJ_UNAUTHORIZED_AID
REJ_INVALID_CERTIFICATE_CHAIN
REJ_INVALID_SIGNATURE
REJ_UNSUPPORTED_MSG_TYPE

5 External functions

The external functions described in table 4 have been defined in order to perform cryptographic operations and handle complex computations.

Table 4: External functions

Function	Parameters			Return	
	Dir.	Name	Type	Value	Type
xf_generateKeyPair	out	p_publicKey	octetstring	Status code	enumerated
	out	p_privateKey	octetstring		
	in	p_algorithm	enumerated		
xf_computeSignature	out	p_signature	octetstring	Status code	enumerated
	in	p_signingKey	octetstring		
	in	p_algorithm	octetstring		
	in	p_data	octetstring		
xf_encryptData	out	p_encryptedData	octetstring	Status code	enumerated
	in	p_encryptionKey	octetstring		
	in	p_algorithm	enumerated		
	in	p_data	octetstring		
xf_generateKey	in	p_algorithm	enumerated	Random Key	octetstring
xf_computeHash	in	p_algorithm	enumerated	Hash	octetstring
	in	p_data	octetstring		
xf_verifySignature	in	p_signature	octetstring	Status code	enumerated
	in	p_verificationKey	octetstring		
	in	p_algorithm	enumerated		
	in	p_data	octetstring		
xf_decryptData	out	p_data	octetstring	Status code	enumerated
	in	p_decryptionKey	octetstring		
	in	p_algorithm	enumerated		
	in	p_encryptedData	octetstring		
xf_verifyGeographicAreas	in	p_containingArea	GeoArea	Is p_containedArea contained in p_containingArea	boolean
	in	p_containedArea	GeoArea		

6 Validity of signed communication

This clause defines the list of actions to be performed by the test system in order to ensure the validity of signed communication.

If possible, every step is handled within TTCN-3, except cryptographical computations. When a cryptographical computation is used, then the name for the external function is provided (e.g. xf_verifySignature). When sending data, TTCN-3 templates used to prepare the messages are expected to be consistent (signer identifier, certificate chain, types, etc.).

6.1 Generating signed data

The test system ensures validity of signed communication by implementing the procedures as described below. This is compliant with IEEE 1609.2 [1], clause 7.2.13.

Inputs:

- private key to be used to sign;
- elliptic curve point format to be used.

NOTE: The certificate is assumed to be consistent with the private key and with the options used to fill in the header fields in the signed data.

Steps:

a) Set all elliptic curve points according to elliptic curve point format

b) Compute hash of encoded data using `xf_computeHash()`

c) Sign the computed hash using `xf_computeSignature()`

6.2 Receiving signed data

The test system ensures validity of signed communication by implementing the procedures as described below. This is compliant with IEEE 1609.2 [1], clause 7.2.19.

Inputs:

- The received secured message.

Steps:

a) Check the internal consistency of the certificate chain:

   1) Signatures using `xf_computeHash()`, `xf_verifySignature()`

   2) Permissions using supersets and matching mechanisms

   3) Geographic scopes using `xf_verifyGeographicAreas()`

b) Check the consistency of the certificate chain with the data. The signed communication may be rejected for any of the following reasons:

   1) "future certificate at generation time"

   2) "expired certificate at generation time"

   3) "expiry date too early"

   4) "expiry date too late"

   5) "signature generated outside certificate validity region" (using `xf_verifyGeographicAreas()`)

   6) "unauthorized PSID" (using supersets and matching mechanisms)

   7) "PSIDs don’t match"

   8) "unauthorized certificate type"

c) Verify the signature:

   1) Compute hash of the data using `xf_computeHash()`

   2) Verify the signature using `xf_verifySignature()`

6.3 Generating enrolment/authorization request

The test system ensures validity of signed communication by implementing the procedures as described below. This is compliant with IEEE 1609.2 [1], clause 7.2.23.

Inputs:

- Private key used to sign the certificate request;

- elliptic curve point format to be used.

NOTE: If the request is signed with a certificate, the certificate is assumed to be consistent with the private key and with the options used to fill in the request.

Steps:

a) Set all elliptic curve points according to elliptic curve point format.

b) Encode ToBeSignedCertificateRequest sub-structure and sign it with private key using `xf_computeSignature()`.

c) Compute the SHA_256 hash of the certificate request using `xf_computeHash()` and set the RequestHash field to the ten least significant bytes of the computed value.

d) Encrypt the certificate request using CA certificate as recipient.

6.4 Receiving enrolment/authorization request

The test system ensures validity of signed communication by implementing the procedures as described below. This is compliant with IEEE 1609.2 [1].

Inputs:

- Received certificate signing request

Steps:

a) Build certificate chain based on certificate information contained in the message.

b) Check the internal consistency of the certificate chain:

   1) Certificate chain has to link to a trusted certificate or be a self-signed certificate

   2) Signatures using `xf_computeHash()`, `xf_verifySignature()`

   3) Permissions using supersets and matching mechanisms

   4) Geographic scopes using `xf_verifyGeographicAreas()`

c) Check the consistency of the certificate chain with the request:

   1) "future certificate at request time"

   2) "expired certificate at request time"

   3) "expiry date too early"

   4) "expiry date too late"

   5) "requested validity region outside signing certificate validity region" (using `xf_verifyGeographicAreas()`)

   6) "unauthorized PSID" (using supersets and matching mechanisms)

   7) "PSIDs don’t match"

   8) "unauthorized certificate type"

d) Verify the signature:

   1) Compute hash of the data using `xf_computeHash()`

   2) Verify the signature using `xf_verifySignature()`

6.5 Generating enrolment/authorization response

The test system ensures validity of signed communication by implementing the procedures as described below. This is compliant with IEEE 1609.2 [1].

Inputs:

- Private key used to sign the certificate
- Data from the request, including certificate fields and response encryption key

NOTE: The CA certificate is assumed to be consistent with the private key and with the options requested by the certificate requester.

Steps:

a) Encode ToBeSignedCertificate sub-structure and sign it with private key using `xf_computeSignature()`

b) Encrypt the certificate response

6.6 Receiving enrolment/authorization response

The test system ensures validity of signed communication by implementing the procedures as described below. This is compliant with IEEE 1609.2 [1], clause 7.2.25.

a) Decrypt EncryptedData

b) Check the internal consistency of the certificate chain:

1) Signatures using `xf_verifySignature()`

2) Permissions using supersets and matching mechanisms

3) Geographic scopes using `xf_verifyGeographicAreas()`

6.7 Data encryption

This is compliant with IEEE 1609.2 [1], clause 7.2.15.

Inputs:

- The message to be encrypted;
- Certificates of all the recipients of the message;
- Symmetric algorithm to be used for encryption.

Steps:

a) For each recipient:

1) Extract the encryption key from the certificate.

b) Generate a random key *k* for the symmetric encryption algorithm using `xf_generateKey()`

c) For each recipient:

1) Extract the public encryption key from the certificate.

2) Encrypt *k* with the public encryption key using `xf_encryptData()`, as a EciesNistP256EncryptedKey.

3) Store the result in the current RecipientInfo sub-structure

d) Encrypt the ToBeEncrypted sub-structure using the symmetric encryption algorithm and the key *k*:

   1) Generate a nonce N of length 12 octets;

   2) Encrypt the data with AES-CCM with inputs nonce N and plaintext ToBeEncrypted sub-structure using `xf_encryptData()` as an AesCcmCiphertext

6.8 Data decryption

This is compliant with IEEE 1609.2 [1], clause 7.8.8.

Inputs:

- Received encrypted message;
- Private key associated to the recipient certificate.

a) Select the relevant RecipientInfo sub-structure in the message

b) Decrypt encrypted symmetric key with the associated asymmetric decryption algorithm using `xf_decryptData()`

c) Decrypt ciphertext using the extracted symmetric key (AES-CCM) using `xf_decryptData()`

7 ATS conventions

The ATS conventions are intended to give a better understanding of the ATS but they also describe the conventions made for the development of the ATS. These conventions shall be considered during any later maintenance or further development of the ATS.

The ATS conventions contain two clauses, the testing conventions and the naming conventions. The testing conventions describe the functional structure of the ATS. The naming conventions describe the structure of the naming of all ATS elements.

To define the ATS, the guidelines of the document ETS 300 406 [9] were considered.

7.1 Testing conventions

7.1.1 Testing states

7.1.1.1 Initial states

7.1.1.1.1 ITS-S send-side states

Depending on preconditions defined in the corresponding test purpose, each testcase starts with one of the following preamble functions to bring IUT to the correct state:

- `f_prNotEnroled()`: ITS-S has all info necessary to send an EnrolmentRequest but does not have any Enrolment credentials yet
- `f_prAwaitingEnrolmentResponse()`: ITS-S has sent an EnrolmentRequest and is waiting for an EnrolmentResponse
- `f_prEnroled()`: ITS-S has received EnrolmentResponse and is able to send AuthorizationRequest
- `f_prAwaitingAuthorizationResponse()`: ITS-S has sent an AuthorizationRequest and is waiting for an AuthorizationResponse
- `f_prAuthorised()`: ITS-S has received a successful AuthorizationResponse

7.1.1.1.2 ITS-S receive-side states

All test cases start with the function `f_prOperationalState()`: ITS-S has the root certificate and is ready to receive messages.

7.1.1.1.3 EA states

All test cases start with the function `f_prOperationalState()`: EA has obtained its certificate and is ready to receive and send Enrolment messages.

7.1.1.1.4 AA states

All test cases start with the function `f_prOperationalState()`: AA has obtained its certificate and is ready to receive and send Authorization messages.

7.1.1.2 Final state

All test cases end with the function f_poDefault. This function brings the IUT back to operational state. As no specific actions are required for the idle state in the base standard, the function f_ poDefault does not invoke any action.

As necessary, further actions may be included in the f_poDefault function.

7.2 Naming conventions

This test suite follows the naming convention guidelines provided in the EG 202 798 [i.1].

7.2.1 General guidelines

The naming convention is based on the following underlying principles:

- in most cases, identifiers should be prefixed with a short alphabetic string (specified in table 5) indicating the type of TTCN-3 element it represents;
- suffixes should not be used except in those specific cases identified in table 7;
- prefixes and suffixes should be separated from the body of the identifier with an underscore ("_");

EXAMPLE 1: `c_sixteen, t_wait`.

- only module names, data type names and module parameters should begin with an upper-case letter. All other names (i.e. the part of the identifier following the prefix) should begin with a lower-case letter;
- the start of second and subsequent words in an identifier should be indicated by capitalizing the first character. Underscores should not be used for this purpose.

EXAMPLE 2: `f_initialState`.

Table 5 specifies the naming guidelines for each element of the TTCN-3 language indicating the recommended prefix, suffixes (if any) and capitalization.

Table 5: ETSI TTCN-3 generic naming conventions

Language element	Naming convention	Prefix	Example Identifier
Module	Use upper-case initial letter	none	IPv6Templates
Group within a module	Use lower-case initial letter	none	messageGroup
Data type	Use upper-case initial letter	none	SetupContents
Message template	Use lower-case initial letter	m_	m_setupInit
Message template with wildcard or matching expression	Use lower-case initial letters	mw_	mw_anyUserReply
Modifying message template	Use lower-case initial letter	md_	md_setupInit
Modifying message template with wildcard or matching expression	Use lower-case initial letters	mdw_	mdw_anyUserReply
Signature template	Use lower-case initial letter	s_	s_callSignature
Port instance	Use lower-case initial letter	none	signallingPort
Test component instance	Use lower-case initial letter	none	userTerminal
Constant	Use lower-case initial letter	c_	c_maxRetransmission
Constant (defined within component type)	Use lower-case initial letter	cc_	cc_minDuration
External constant	Use lower-case initial letter	cx_	cx_macId
Function	Use lower-case initial letter	f_	f_authentication()
External function	Use lower-case initial letter	fx_	fx_calculateLength()
Altstep (incl. Default)	Use lower-case initial letter	a_	a_receiveSetup()
Test case	Use ETSI numbering	TC_	TC_COR_0009_47_ND
Variable (local)	Use lower-case initial letter	v_	v_macId
Variable (defined within a component type)	Use lower-case initial letters	vc_	vc_systemName
Timer (local)	Use lower-case initial letter	t_	t_wait
Timer (defined within a component)	Use lower-case initial letters	tc_	tc_authMin
Module parameters for PICS	Use all upper case letters	PICS_	PICS_DOOROPEN
Module parameters for other parameters	Use all upper case letters	PX_	PX_TESTER_STATION_ID
Formal Parameters	Use lower-case initial letter	p_	p_macId
Enumerated Values	Use lower-case initial letter	e_	e_syncOk

7.2.2 ITS specific TTCN-3 naming conventions

Next to such general naming conventions, table 6 shows specific naming conventions that apply to the ITS TTCN-3 test suite.

Table 6: ITS specific TTCN-3 naming conventions

Language element	Naming convention	Prefix	Example Identifier
ITS Module	Use upper-case initial letter	Its"IUTname"_	ItsSecurity_
Module containing types and values	Use upper-case initial letter	Its"IUTname"_TypesAndValues	ItsSecurity_TypesAndValues
Module containing Templates	Use upper-case initial letter	Its"IUTname"_Templates	ItsSecurity_Templates
Module containing test cases	Use upper-case initial letter	Its"IUTname"_TestCases	ItsSecurity_TestCases
Module containing functions	Use upper-case initial letter	Its"IUTname"_Functions	ItsSecurity_Functions
Module containing external functions	Use upper-case initial letter	Its"IUTname"_ExternalFunctions	ItsSecurity_ExternalFunctions
Module containing components, ports and message definitions	Use upper-case initial letter	Its"IUTname"_Interface	ItsSecurity_Interface
Module containing main component definitions	Use upper-case initial letter	Its"IUTname"_TestSystem	ItsSecurity_TestSystem
Module containing the control part	Use upper-case initial letter	Its"IUTname"_TestControl	ItsSecurity_TestControl

7.2.3 Usage of Log statements

All TTCN-3 log statements use the following format using the same order:

- Three asterisks.
- The TTCN-3 test case or function identifier in which the log statement is defined.
- One of the categories of log: INFO, WARNING, ERROR, PASS, FAIL, INCONC, TIMEOUT.
- Free text.
- Three asterisks.

EXAMPLE 1:

```
log("*** TP_SEC_ITS-S_ENR_NB_06: INFO: Preamble: Received and answered
Enrolment Request ***");
```

Furthermore, the following rules are applied for the ITS-Security ATS:

- Log statements are used in the body of the functions, so that invocation of functions are visible in the test logs.
- All TTCN-3 setverdict statement are combined (as defined in TTCN-3 v3.4.1) with a log statement following the same above rules (see example below).

EXAMPLE 2:

```
setverdict(pass, "*** TP_SEC_ITS-S_ENR_NB_06: PASS: Enrolment Response
correctly accepted ***");
```

7.2.4 Test Case (TC) identifier

The table 7 shows the test case naming convention, which follows the same naming convention as the test purposes.

Table 7: TC naming convention

Identifier:	TC_<root>_<gr>_<sgr>_<x>_<nn>		
	<root> = root	SEC	
	<gr> = group	CA	Certificate Authorithy
		EA	Enrolment Authorithy
		AA	Authorization Authority
		ITS-S	ITS Station
	<sgr> =sub-group	ENR	Enrolment
		AUTH	Autorisation
		S-DATA	Send Data
		R-DATA	Receive Data
	<x> = type of testing	NB	Normal Behaviour
		EB	Exceptional Behaviour
	<nn> = sequential number		01 to 99
	<X> = Variant for 1st permutation table		A to Z
	<Y> = Variant for 2nd permutation table		A to Z

EXAMPLE: TP identifier: TP/SEC/ITS-S/ENR/NB-02
TC identifier: TC_SEC_ITS-S_ENR_NB_02

7.3 On line documentation

Using the T3D tool enables providing on-line documentation browser in HTML, by tagging TTCN-3 comments. These tags are defined in table 8.

Table 8: TTCN-3 comment tags

Tag	Description
@author	Specifies the names of the authors or an authoring organization which either has created or is maintaining a particular piece of TTCN-3 code.
@desc	Describes the purpose of a particular piece of TTCN-3 code. The description should be concise yet informative and describe the function and use of the construct.
@remark	Adds extra information, such as the highlighting of a particular feature or aspect not covered in the description.
@see	Refers to other TTCN-3 definitions in the same or another module.
@return	Provides additional information on the value returned by a given function.
@param	Documents the parameters of parameterized TTCN-3 definitions.
@version	States the version of a particular piece of TTCN-3 code.

The HTML files result from the compilation of the TTCN-3 modules with the T3D tool. These HTML files are ready for browsing, and contain links enabling to navigate through the ATS.

EXAMPLE:

```
/**
 * @desc  Check that ITS-S generates valid enrolment request
 *        with a different response_encryption_key for every request
 * @version 0.0.11
 * @see   Draft ETSI TS 103 096-2 TP/SEC/ITS-S/ENR/NB-06
 */
```

Annex A (informative): ATS in TTCN-3

A.1 TTCN-3 files and other related modules

Void

A.2 HTML documentation of TTCN-3 files

Void

Annex B (normative): Partial PIXIT proforma for Security

The PIXIT Proforma is based on ISO/IEC 9646-6 [7]. Any needed additional information can be found in this international standard document.

B.1 Identification summary

Table B.1

PIXIT Number:	
Test Laboratory Name:	
Date of Issue:	
Issued to:	

B.2 ATS summary

Table B.2

Protocol Specification:	TS 102 941
Protocol to be tested:	Trust and Privacy Management
ATS Specification:	TS 103 096-3
Abstract Test Method:	Clause 4

B.3 Test laboratory

Table B.3

Test Laboratory Identification:	
Test Laboratory Manager:	
Means of Testing:	
SAP Address:	

B.4 Client identification

Table B.4

Client Identification:	
Client Test manager:	
Test Facilities required:	

B.5 SUT

Table B.5

Name:	
Version:	
SCS Number:	
Machine configuration:	
Operating System Identification:	
IUT Identification:	
PICS Reference for IUT:	
Limitations of the SUT:	
Environmental Conditions:	

B.6 Protocol layer information

B.6.1 Protocol identification

Table B.6

Name:	TS 102 941
Version:	
PICS References:	TS 103 096-1

B.6.2 IUT information

Void

Annex C (normative): PCTR Proforma for Security

The PCTR proforma is based on ISO/IEC 9646-6 [7]. Any needed additional information can be found in this International standard document.

C.1 Identification summary

C.1.1 Protocol conformance test report

Table C.1

PCTR Number:	
PCTR Date:	
Corresponding SCTR Number:	
Corresponding SCTR Date:	
Test Laboratory Identification:	
Test Laboratory Manager:	
Signature:	

C.1.2 IUT identification

Table C.2

Name:	
Version:	
Protocol specification:	
PICS:	
Previous PCTR if any:	

C.1.3 Testing environment

Table C.3

PIXIT Number:	
ATS Specification:	
Abstract Test Method:	
Means of Testing identification:	
Date of testing:	
Conformance Log reference(s):	
Retention Date for Log reference(s):	

C.1.4 Limits and reservation

Additional information relevant to the technical contents or further use of the test report, or the rights and obligations of the test laboratory and the client, may be given here. Such information may include restriction on the publication of the report.

.....

.....

.....

.....

C.1.5 Comments

Additional comments may be given by either the client or the test laboratory on any of the contents of the PCTR, for example, to note disagreement between the two parties.

.....

.....

.....

.....

C.2 IUT Conformance status

This IUT has or has not been shown by conformance assessment to be non-conforming to the specified protocol specification.

Strike the appropriate words in this sentence. If the PICS for this IUT is consistent with the static conformance requirements (as specified in clause C.3 in this report) and there are no "FAIL" verdicts to be recorded (in clause C.6 in this report) strike the words "has or", otherwise strike the words "or has not".

C.3 Static conformance summary

The PICS for this IUT is or is not consistent with the static conformance requirements in the specified protocol.

Strike the appropriate words in this sentence.

C.4 Dynamic conformance summary

The test campaign did or did not reveal errors in the IUT.

Strike the appropriate words in this sentence. If there are no "FAIL" verdicts to be recorded (in clause C.6 of this report) strike the words "did or" otherwise strike the words "or did not".

Summary of the results of groups of test:

.....

.....

.....

.....

.....

.....

.....

.....

C.5 Static conformance review report

If clause C.3 indicates non-conformance, this clause itemizes the mismatches between the PICS and the static conformance requirements of the specified protocol specification.

.....

.....

.....

.....

.....

.....

.....

.....

C.6 Test campaign report

Table C.4: Test Cases

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause C.7)
TC_SEC_ITS-S_ENR_NB_01	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_02	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_04	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_05	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_06	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_07	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_10	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_11	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_12	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_13	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_16	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_17	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_18	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_20	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_21	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_NB_22	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_01	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_02	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_03	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_04	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_05	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_06	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_07	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_08	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_09	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_10	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_11	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_12	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_13	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_14	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_15	Yes/No	Yes/No		
TC_SEC_ITS-S_ENR_EB_16	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_01	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_02	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_03	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_06	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_07	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_08	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_09	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_10	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_20	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_21	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_23	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_24	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_25	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_30	Yes/No	Yes/No		
TC_SEC_EA_ENR_NB_32	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_01	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_02	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_03	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_04	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_05	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_06	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_07	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_08	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_09	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_10	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause C.7)
TC_SEC_ITS-S_AUTH_NB_11	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_12	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_13	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_14	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_15	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_16	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_17	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_18	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_19	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_20	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_NB_21	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_01	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_02	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_03	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_04	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_05	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_06	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_07	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_08	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_09	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_10	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_11	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_12	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_13	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_14	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_15	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_16	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_17	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_18	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_19	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_20	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_21	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_22	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_23	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_24	Yes/No	Yes/No		
TC_SEC_ITS-S_AUTH_EB_25	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_01	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_02	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_03	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_04	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_04a	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_04b	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_06	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_05	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_07	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_08	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_09	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_10	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_11	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_12	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_13	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_14	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_15	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_19	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_20	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_21	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_22	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_23	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_24	Yes/No	Yes/No		
TC_SEC_AA_AUTH_NB_25	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_01	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_02	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause C.7)
TC_SEC_AA_AUTH_EB_03	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_04	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_05	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_06	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_07	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_08	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_09	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_10	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_11	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_12	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_13	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_14	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_15	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_16	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_17	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_18	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_19	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_20	Yes/No	Yes/No		
TC_SEC_AA_AUTH_EB_21	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_01	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_02	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_03	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_04	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_05	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_06	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_07	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_08	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_09	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_10	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_11	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_12	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_13	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_14	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_15	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_16	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_17	Yes/No	Yes/No		
TC_SEC_ITS-S_S-DATA_NB_18	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_01	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_02	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_03	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_04	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_05	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_06	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_07	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_NB_08	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_02	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_03	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_04	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_05	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_06	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_07a	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_07b	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_07c	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_08	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_09	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_10	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_11	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_12	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_13	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_14	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_15	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_16	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause C.7)
TC_SEC_ITS-S_R-DATA_EB_17	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_18	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_19	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_20	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_21	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_22	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_23	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_24	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_25	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_26	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_27	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_28	Yes/No	Yes/No		
TC_SEC_ITS-S_R-DATA_EB_29	Yes/No	Yes/No		

C.7 Observations

Additional information relevant to the technical content of the PCTR is given here.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

History

Document history		
V1.1.1	July 2013	Publication